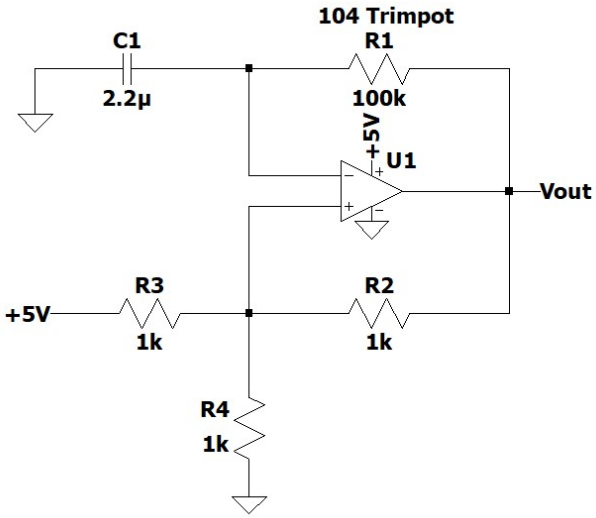
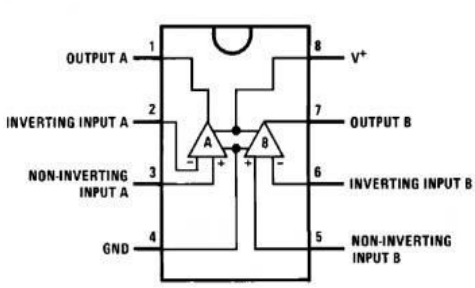
ASSIGNMENT – 3

1. **Build a square wave oscillator with given op-amps; 50% duty cycle and variable frequency 100 Hz to 1k Hz.**



(EXPECTED OUTPUT)

OP-AMP used is LM358L.



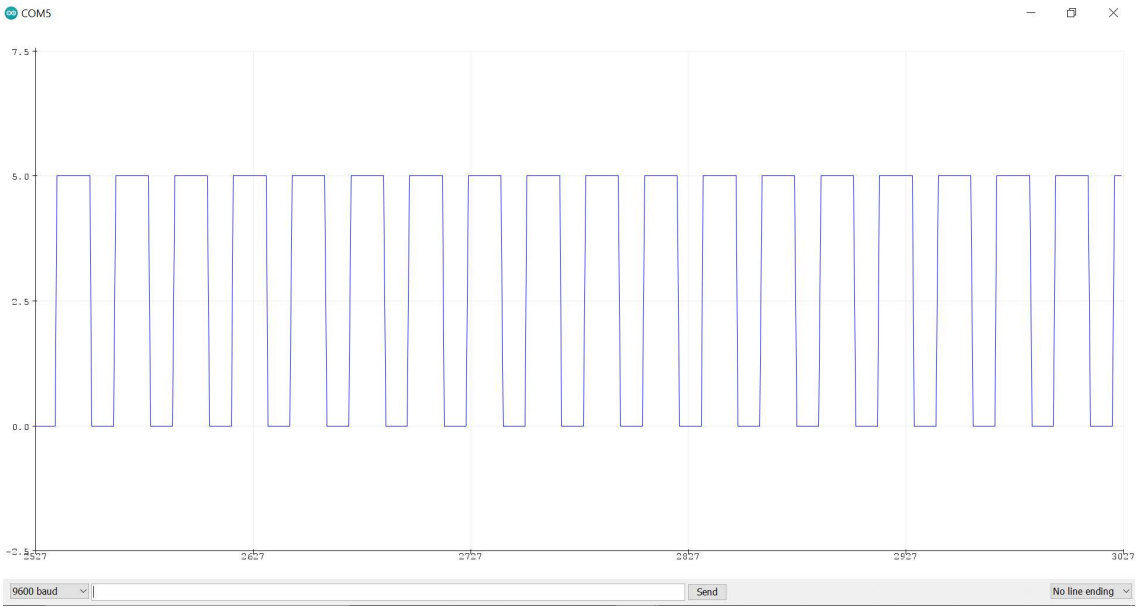
Built- in op-amp A is used for square wave generation and op-amp B is used for sine wave generation (from square wave).

Resistor R3 is connected to +5V so that capacitor C1 gets enough time to get charged and discharged. Though we are required to get 50% duty cycle it can't be achieved with the current set-up and can be tinkered with the help of code or additional hardware to achieve it.

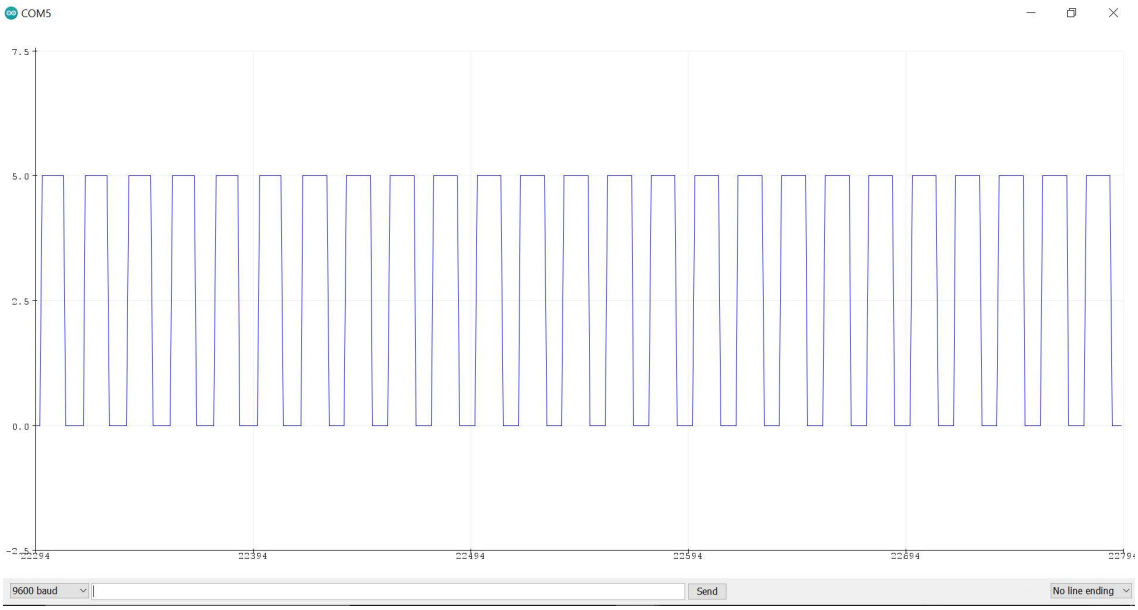
The Arduino code for the set-up is

```
float vol1 = 0;
float data1 = 0;
const int in = 6;

void setup()
{
  pinMode(in, INPUT);
  pinMode(13, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  data1 = digitalRead(in);
  vol1 = 5 * data1;
  Serial.println(vol1);
  delay(10);
}
```

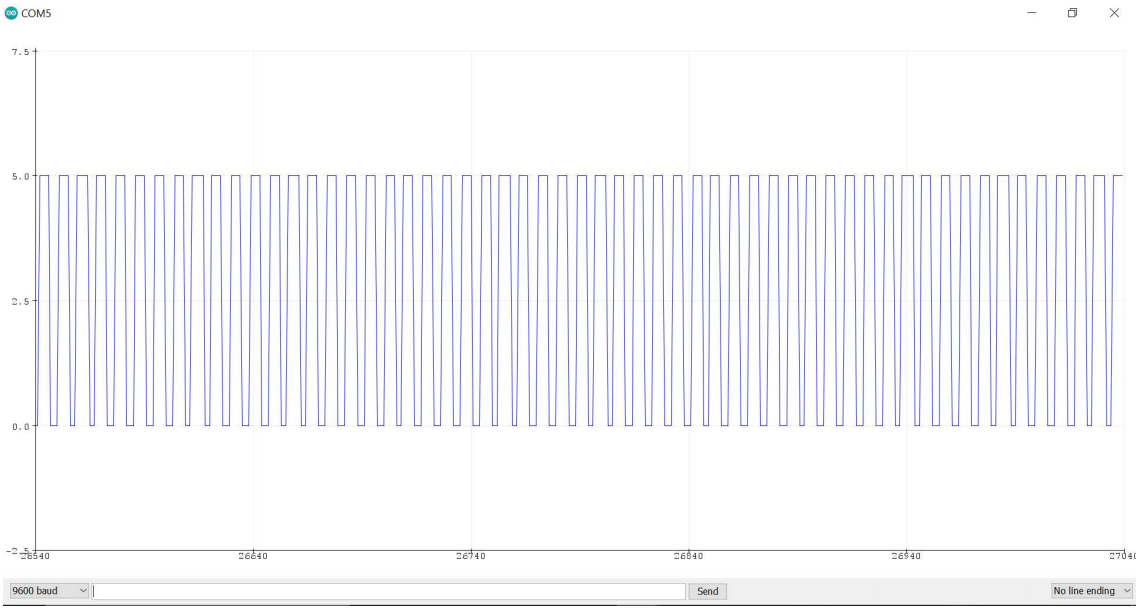
The output of the above set-up is as follows:



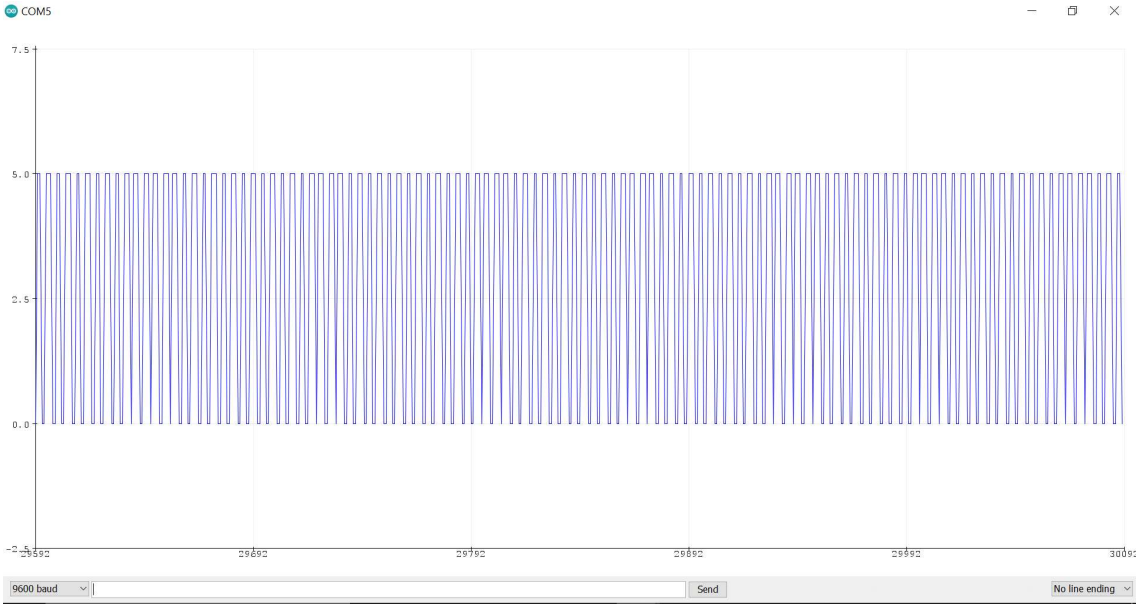
32.5 Hz



50 Hz



110 Hz



230 Hz

(Frequency mentioned above are square wave frequency)

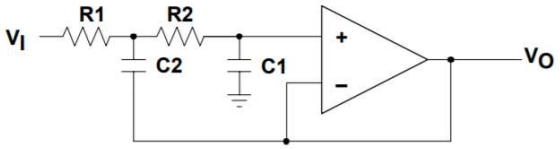
To measure the duty cycle of the square, we use a Digital Multimeter to measure the voltage across the output of LM358L and ground (gives average over a time period). On measuring, we get, value supplied to the circuit is **4.98** and average voltage of the square value over a time period as **2.72**. So, we got a square wave of duty cycle approx. **55%**.

To get exactly 50% duty cycle, we can change the code so that it takes one rising edge to the other as one high or low condition or use flip-flops.

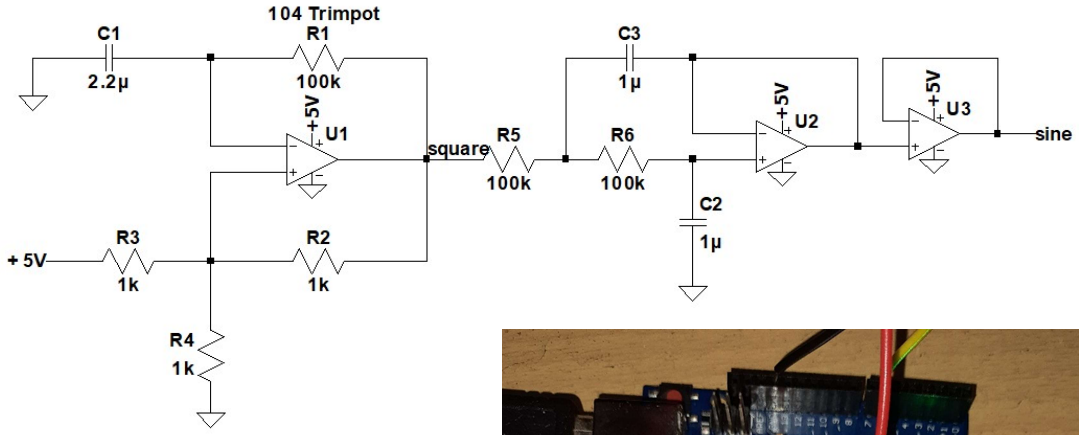
(I couldn't get the exact value so I skipped mentioning it.)

2. Design an LPF to get a sine wave with the variable frequency from the output of the square wave generator and buffer it to drive the load.

We use Sallen key low pass filter to filter the square wave to a sine wave. The circuit diagram for the Sallen key LPF is as below:



Here, V_I is square wave generated from the previous question and Vo is the sine wave output which is further processed using a buffer to a desired output. Full set-up is

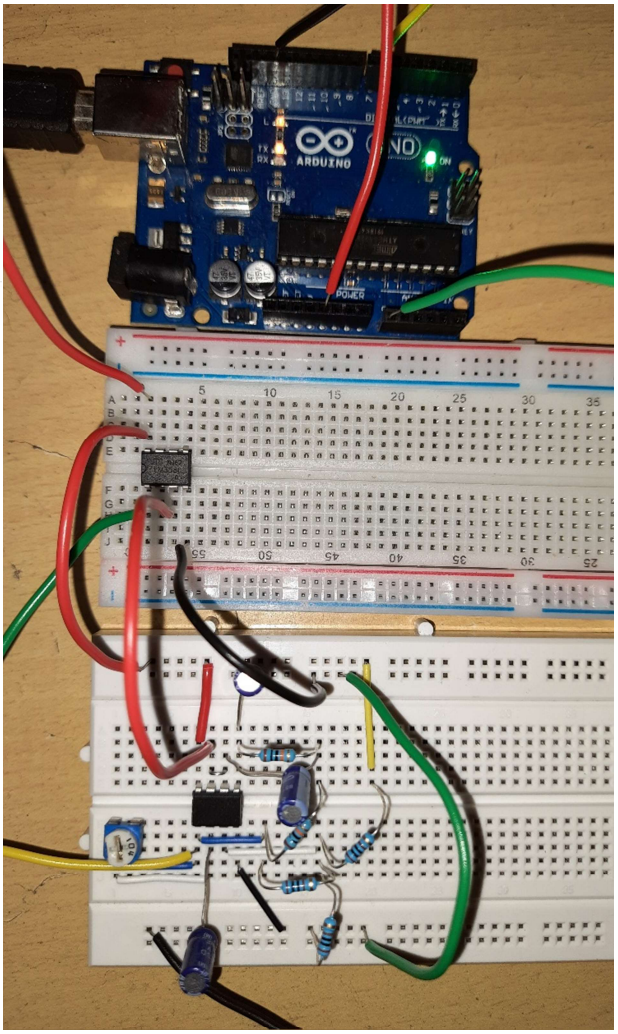


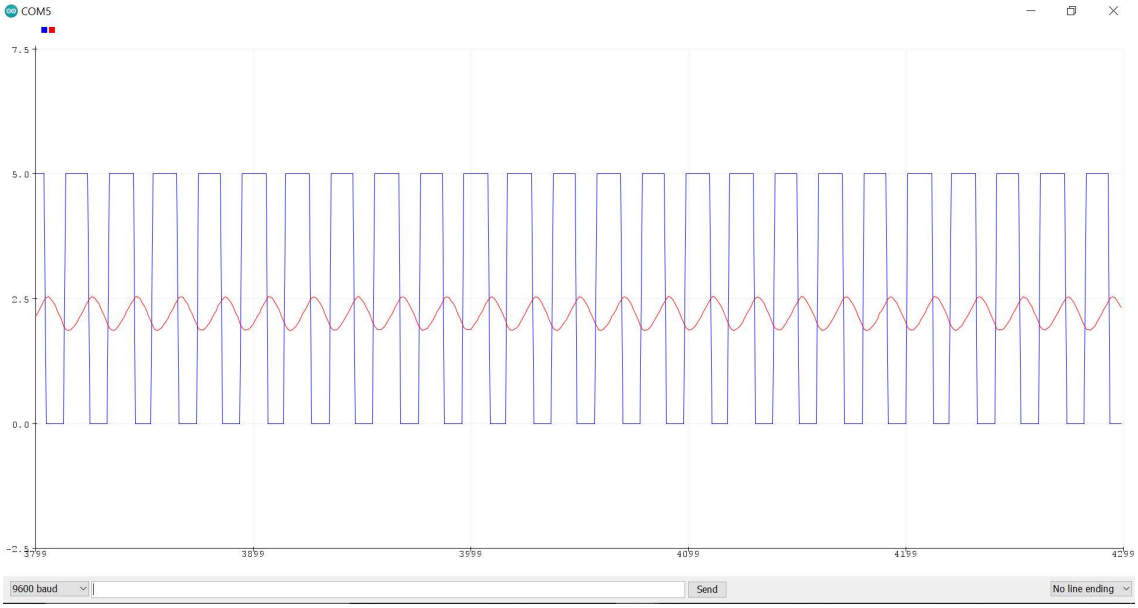
Output at square:

Output at sine :

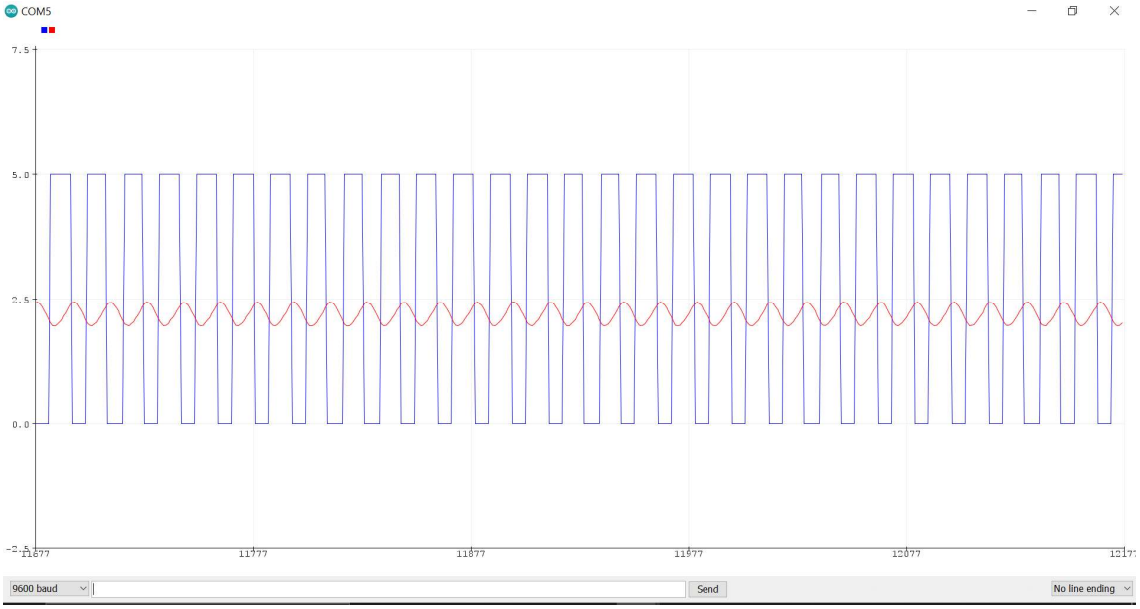
The Arduino code final code is

```
float vol1 = 0;
float vol2 = 0;
float data1 = 0;
float data2 = 0;

void setup()
{
 pinMode(13, OUTPUT);
 pinMode(6, INPUT);
 Serial.begin(9600);

void loop()
{
 data1 = digitalRead(6);
 vol1 = 5 * data1 ;
 Serial.println(vol1);
 data2 = analogRead(A0);
 vol2 = 5 * data2 / 1023.0 ;
 Serial.print(",");
 Serial.println(vol2);
 delay(10);
}
```

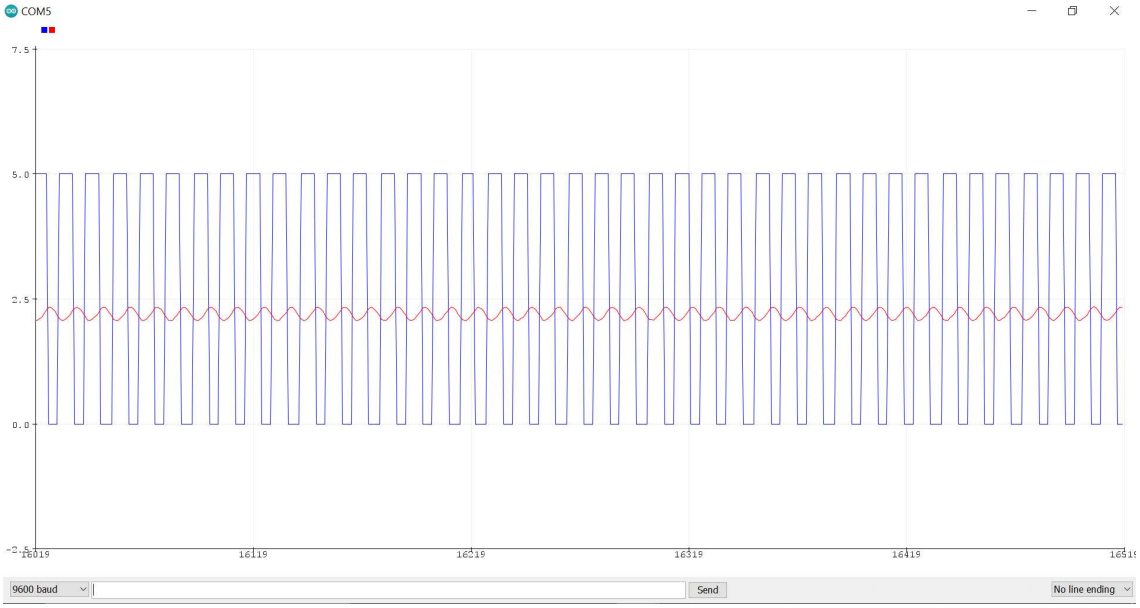



COM5
7.5
5.0
2.5
0.0
-2.5
3799
3899
3999
4099
4199
4299
9600 baud
Send
No line ending

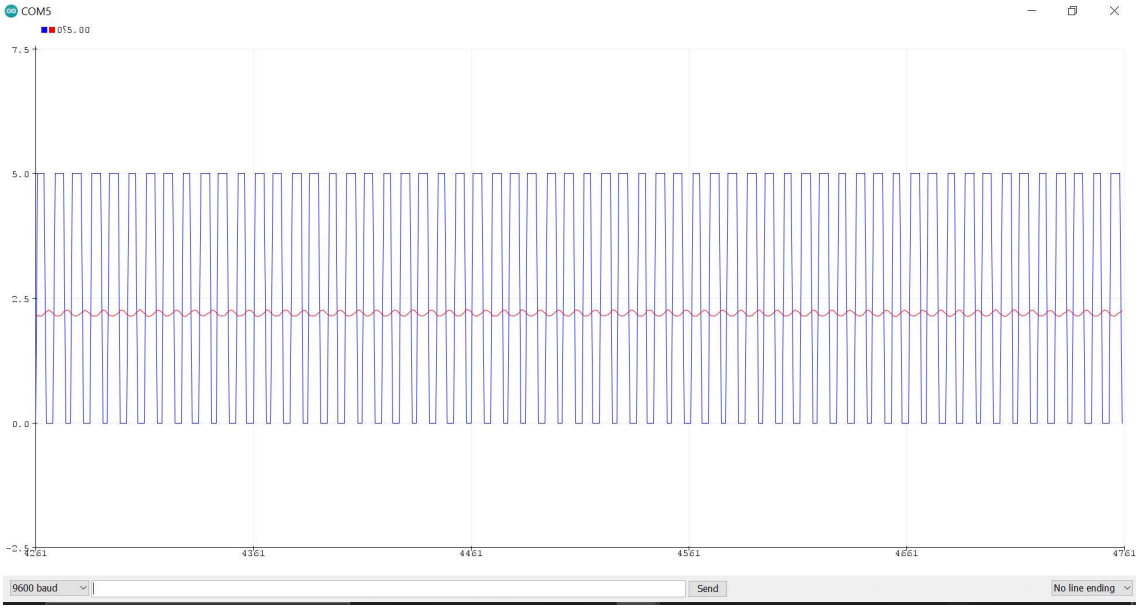
22.5 Hz


COM5
7.5
5.0
2.5
0.0
-2.5
11677
11777
11877
11977
12077
9600 baud
Send
No line ending

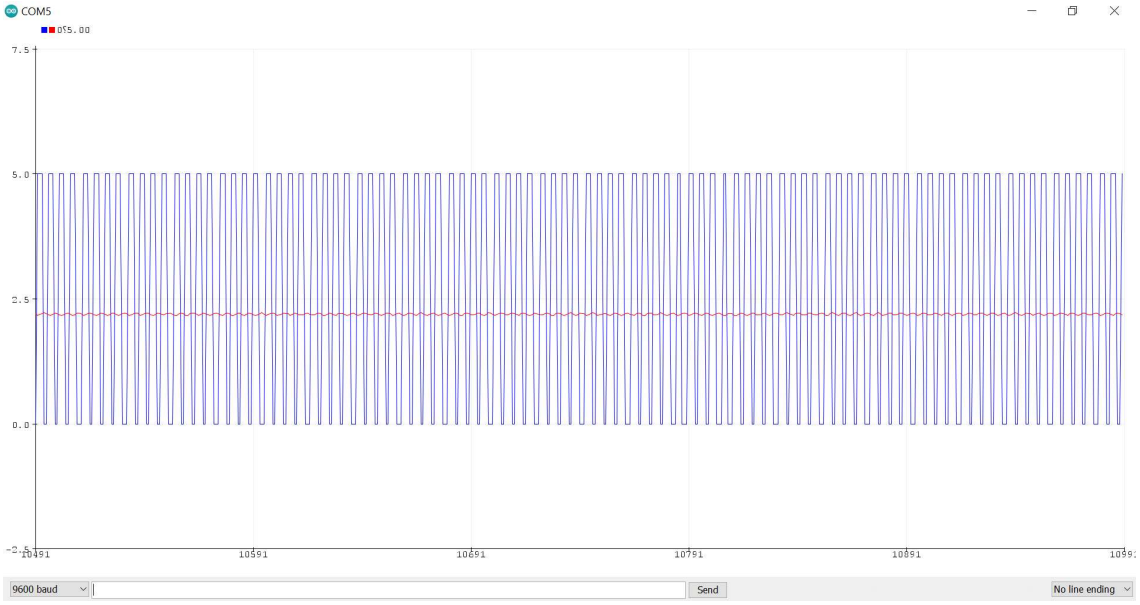
30 Hz


COM5
7.5
5.0
2.5
0.0
-2.5
16019
16119
16219
16319
16419
9600 baud
Send
No line ending

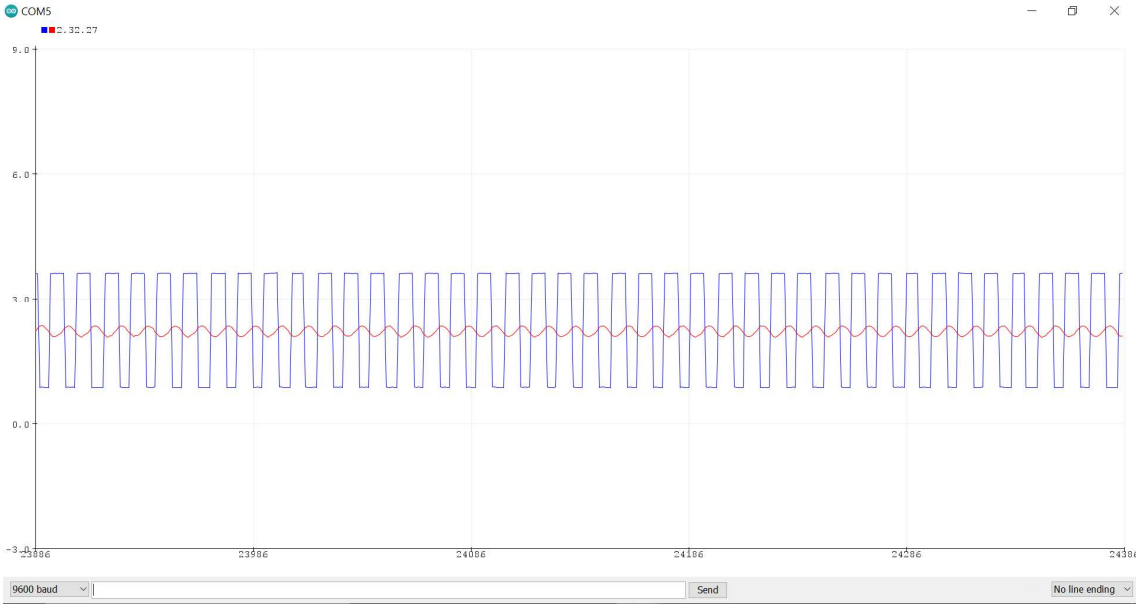
37.5 Hz


COM5
055.00
7.5
5.0
2.5
0.0
-2.5
4261
4361
4461
4561
4661
4761
9600 baud
Send
No line ending

60 Hz

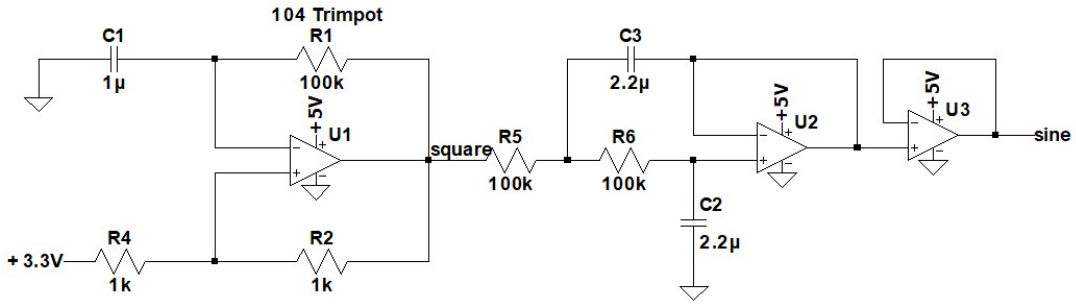


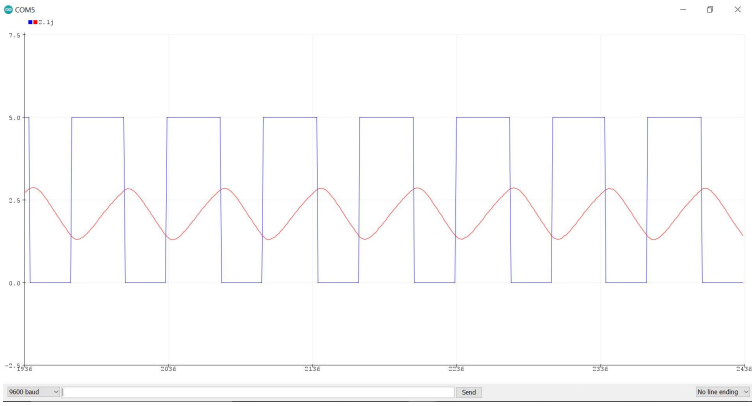
96.25 Hz



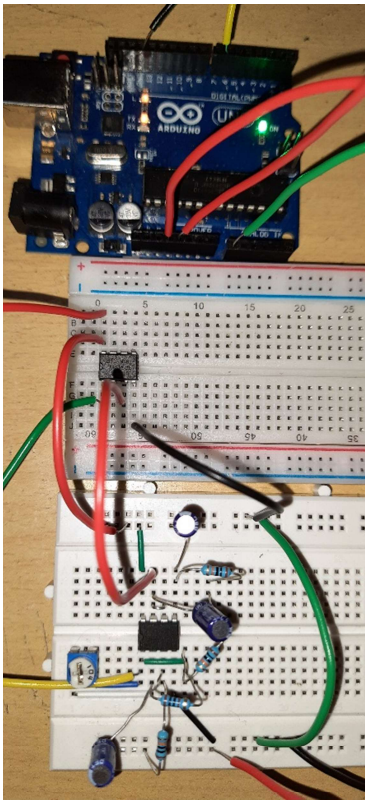
Output when taking square from A0 and sine from A1 (37.5 Hz)
(Frequency mentioned above are sine wave frequency)

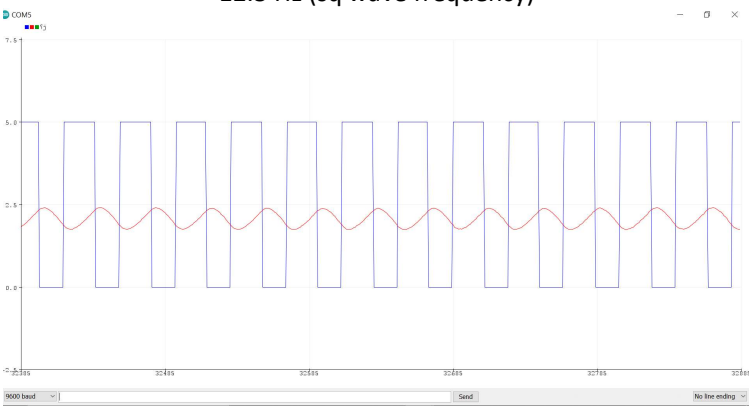
There are many ways to generate square wave and sine wave. Another notable way to generate them is using the below set-up and same Arduino code as above set-up.



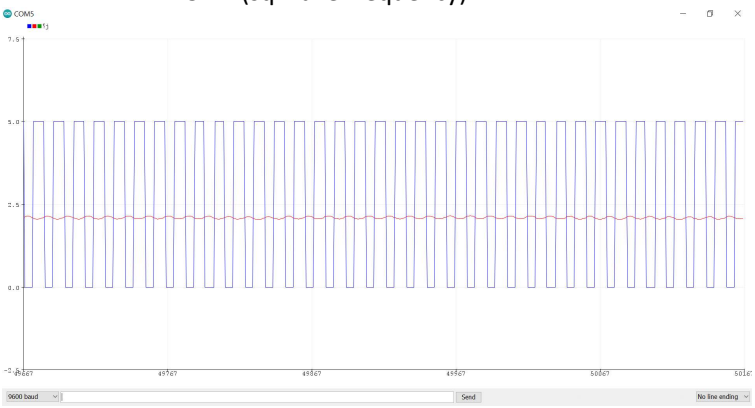


12.5 Hz (sq wave frequency)

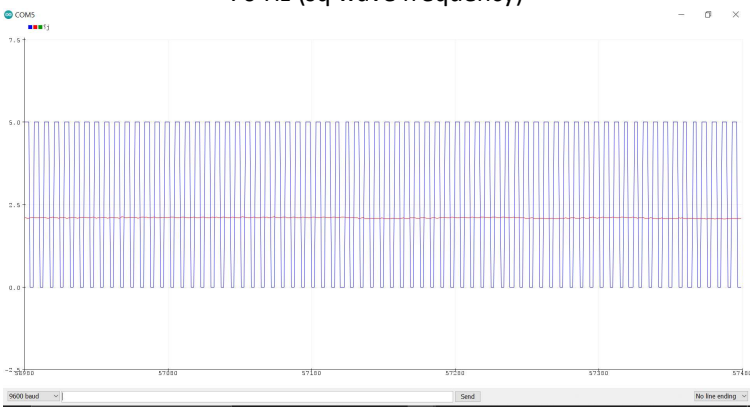




25 Hz (sq wave frequency)



70 Hz (sq wave frequency)



135 Hz (sq wave frequency)